## linha horizontal



PHP

04/09/2024

**─**

Control Structures, Object-Oriented Programming and Database Interface in PHP

Knowledge Evidence Quiz

# 

# Overview

This document provides an overview of essential PHP programming concepts, focusing on control structures, object-oriented programming (OOP), and database interfaces. It begins by examining the three primary control structures: selection, iteration, and sequence, that guide the flow of a PHP script, ensuring it performs tasks based on specific conditions. The discussion then shifts to OOP, emphasizing its role in creating modular, maintainable code by encapsulating data and behavior within objects, which is crucial for effective web development. Lastly, the document explores database interfaces, particularly using PHP’s mysqli extension to interact with MySQL databases, illustrating how PHP facilitates efficient data management in web applications. Practical examples are included to demonstrate the application of these concepts in real-world scenarios.

# 

# 

# 

# 

# 

# Programming Control Structures in PHP

There are three kind of control structures in PHP. These are:

1. **Conditional**: allow the execution of specific blocks of code based on the evaluation of a condition or Boolean expression. These structures are crucial for making decisions within a script. The most commonly used conditional structures in PHP are if, if-else, and switch.

| $number = 10;  if ($number > 5) {  echo "The number is greater than 5."; } // Output: The number is greater than 5. |
| --- |

| $number = 3;  if ($number > 5) {  echo "The number is greater than 5."; } else {  echo "The number is not greater than 5."; } // Output: The number is not greater than 5. |
| --- |

| $day = "Monday";  switch ($day) {  case "Monday":  echo "Start of the work week!";  break;  case "Friday":  echo "Almost the weekend!";  break;  default:  echo "Just another day.";  break; } // Output: Start of the work week! |
| --- |

1. **Iteration**: Iteration structures in PHP allow for the repeated execution of code blocks, useful for tasks that need repetition, such as looping through arrays or generating repeated outputs. The two main types are while and do-while loops.

| // while Loop: Executes the code block as long as the condition is true. If the condition is false initially, the loop won't run.  $count = 1;  while ($count <= 5) {  echo "Count is: $count <br>";  $count++; } |
| --- |

| // do-while Loop: Executes the code block at least once before checking the condition, ensuring the loop runs at least one time.  $count = 1;  do {  echo "Count is: $count <br>";  $count++; } while ($count <= 5); |
| --- |

1. **Sequence:** is the simplest control structure, where instructions are executed in the order they appear, one after another. This linear flow ensures that each statement is processed sequentially without any condition or iteration affecting the sequence.

| $numbers = range(1, 5);  foreach ($numbers as $number) {  echo "Number: $number <br>"; } |
| --- |

# Object-Oriented Programming (OOP) in Web Development

Object-Oriented Programming (OOP) is a paradigm that organizes software design around objects in data structures that contain properties and methods. In web development, OOP enhances code modularity, reusability, and maintainability by encapsulating data and behavior within classes. For example, consider a Fruit class:

| class Fruit {  public $name;  public $color;   function \_\_construct($name, $color) {  $this->name = $name;  $this->color = $color;  }   function displayInfo() {  echo "This is a $this->color $this->name.";  } }  $apple = new Fruit("Apple", "red"); $apple->displayInfo(); // Output: This is a red Apple. |
| --- |

## 

# 

# Database Interface

A database interface in PHP means to interact with a database, allowing you to perform operations such as querying, updating, and managing data. The mysqli extension is commonly used for connecting to MySQL databases, making it easy to execute SQL commands and retrieve data. This interaction is crucial in web development, where dynamic content often using on data stored in databases. Check bellow example:

| <?php // Create database connection $mysqli = new mysqli("localhost", "root", "qwer1234!", "products");  // Check connection if ($mysqli->connect\_error) {  die("ERROR CONNECTION DATABASE" . $mysqli->connect\_error); }  // Query to retrieve products $sql = "SELECT id, name, price FROM products"; $result = $mysqli->query($sql);  // Check if results are returned if ($result->num\_rows > 0) {  // Output data of each row  while($row = $result->fetch\_assoc()) {  echo "Product ID: " . $row["id"] . " - Name: " . $row["name"] . " - Price: $" . $row["price"] . "<br>";  } } else {  echo "No products found."; }  // Close connection $mysqli->close(); ?> |
| --- |